REMARKS

In response to the Office Action mailed July 25, 2008, Applicants respectfully request reconsideration. Claims 1-11 were previously pending in this application. By this amendment, claims 1-9 and 11 have been amended. New claims 12-15 have been added. As a result, claims 1-15 are pending for examination with claims 1, 7, 8 and 11 being independent. No new matter has been added.

Rejections under 35 U.S.C. §112

The Office Action rejected claim 9 under 35 U.S.C. §112, second paragraph, as purportedly being indefinite. Applicants have amended claim 9 to address the Examiner's concerns.

Accordingly, withdrawal of this rejection is respectfully requested.

Rejections Under 35 U.S.C. §103

The Office Action rejected claims 1-11 under 35 U.S.C. 103(a) as being allegedly unpatentable over Nexus 5001 Forum: Standard for a Global Embedded Processor Debug Interface (The Nexus 5001 Forum), hereinafter "Nexus," in view of Argade et al., (U.S. Patent No. 5,724,505), hereinafter "Argade." Applicants respectfully disagree.

A. Independent Claim 1

Claim 1, as amended, recites:

> A method for transmitting digital messages on execution of an instruction sequence by a microprocessor, through output terminals of a monitoring circuit integrated on the microprocessor, at least one digital message of said digital messages being representative of characteristic data stored by the monitoring circuit on detection of a jump in the execution of an instruction sequence from an initial instruction to a destination instruction different from an instruction following the initial instruction in the instruction sequence, the method comprising steps of:
>
> determining whether the jump is associated with a jump instruction explicitly indicating an address of a destination instruction of the jump;
>
> when it is determined that that the address of the destination instruction is explicitly indicated in the jump instruction:
>
> assigning a first value to a first set of bits of at least one digital message to provide an explicit jump message, and

transmitting the explicit jump message; and

when it is determined that the address of the destination instruction is not explicitly indicated in the jump instruction:

assigning a second value to the first set of bits of at least one digital message to provide an implicit jump message indicating an occurrence of an implicit jump,

adding a field to the implicit jump message, the field comprising a second set of bits identifying a type of the implicit jump from among several types of implicit jumps, *wherein the field is added only when the address of the destination instruction is not explicitly indicated in the jump instruction*, and

transmitting the implicit jump message.

(Emphasis added).

On page 4, the Office Action concedes that Nexus "does not explicitly disclose when the first set of bits is at the second value, providing an additional field in the implicit jump message, the additional field comprising a second set of bits, and assigning to the second set of bits a third value identifying the jump as an implicit jump from among several types of implicit jumps." The Office Action then states that Argade "does disclose of identifying a jump as an implicit jump from among several types of implicit jumps by assigning to a set of bits a specific value (col. 5, lines 39-45, the INSTR_TYPE, which is part of a digital message of lines 24-27, with the types of jumps being type_1, type_2, and type_3 in col. 5, lines 49-67)."

Applicants respectfully note that Argade discusses providing INSTR_TYPE signals for three types of discontinuities. The discontinuities include the "type-3" discontinuity that "is preferably a program counter relative or absolute address instruction, including, but not being limited to, a program counter relative or absolute address jump or call" for which "it is not necessary to record an address at all, because both the origin and the DEST_ADDR may be easily determined from the program listing" (col. 5, lines 65-67 – col. 4, lines). Argade describes that the discontinuity and conditionally executed instructions are received as INSTR_TYPE signals from the processor core 12 which indicate the type of **each** instruction executed (col. 5, lines 39-41) (emphasis added). In contrast, claim 1, as amended, recites adding a field to the implicit jump message, the field comprising a second set of bits identifying a type of the implicit jump from among several types of implicit jumps, *wherein the field is added only when the address of the destination instruction is not explicitly indicated in the jump instruction* (emphasis

added). Support for this amendment can be found at least on page 8, lines 26-29 of Applicants' specification. Argade does not teach or suggest the above limitation of claim 1.

Further, on page 6, the Office Action states that it would have been readily recognized to one of ordinary skill in the art at the time of the invention that the teaching of *a jump type field of* Argade could be additionally added to the program trace, *indirect branch message* of Nexus 5001 Forum, Table 6-7 *without also adding it to the program trace, direct branch message* of Nexus 5001 Forum, Table 6-6 (emphasis added). Applicants respectfully note that Argade does not discuss a jump type **field** as stated in the Office Action (emphasis added). In fact, nowhere in the reference does Argade even mention a field. Instead, Argade discusses INSTR_TYPE *signals* provided over an instruction-type line (emphasis added). Argade discusses that, by identifying an INSTR_TYPE as one of the three types of pre-defined discontinuities, the TBC block 50 may determine whether to record or to discard its corresponding address (or addresses) and whether additional information about the INSTR_TYPE needs to be recorded (col. 5, lines 43-48).

On page 4, the Office Action states that "Argade's teaching of a field which identifies the jump from among several types of jumps enables the capturing of information about whether certain types of instructions were executed, including conditionally executed instructions; a trace of these instructions may be very important in debugging most programs (Argade, col. 3, lines 17-21)." In the cited passage, Argade discusses that "the discontinuity buffer does not provide a trace of *conditionally executed instructions*, such as "IF/THEN/ELSE" instructions because they are *not true discontinuities*. As a result, the limited trace contains no information about whether these instructions were executed. A trace of these instructions may be very important in debugging most programs" (Argade, col. 3, lines 17-23) (emphasis added). Thus, in this portion, Argade discusses that a trace of *conditionally executed instructions*, such as "IF/THEN/ELSE" instructions, may be very important in debugging most programs (emphasis added).

Argade states that conditionally executed instructions, such as if-then-else, are not discontinuities per se, but may be treated as a type-3 discontinuity (Argade, col. 6, lines 6-9). For the type-3 discontinuities only an indicator of whether the type-3 discontinuity was actually executed is required (Argade, col. 6, lines 4-6).

In view of the above, it is not clear why one of skill in the art would add an INSTR_TYPE signal Argade could be added to the program trace, indirect branch message of

Nexus 5001 Forum, Table 6-7 *without also adding it to the program trace, direct branch message of Nexus* 5001 Forum, Table 6-6, as suggested in the Office Action (emphasis added). Nowhere does Argade suggest that an INSTR_TYPE signal may not be provided for a "type-3" discontinuity which is, according to the Office Action "fits into Nexus 5001 Forum's teaching of his program trace, direct branch message of Table 6-6, which does not have an address field." On the contrary, as discussed above, Argade specifically describes that the discontinuity and conditionally executed instructions are received as INSTR_TYPE signals from the processor core 12 which indicate *the type of each instruction executed* (col. 5, lines 39-41) (emphasis added). INSTR_TYPE signals for the "type_3" discontinuities including conditionally executed instructions are received from the processor as part of Argade's improvement over systems where "the discontinuity buffer does not provide a trace of conditionally executed instructions" (Argade, col. 3, lines 17-18). In fact, the trace recording hardware of Argade *stores the instruction type* and, optionally, data indicative of whether the instruction type was actually executed, in the instruction type FIFO, and stores its associated abbreviated address in the address FIFO (Argade, col. 3, lines 58-62) (emphasis added).

On page 6, the Office Action states that "Nexus 5001 Forum, Tables 6-6 and 6-7, shows that the two different types of messages *are already of different length*; it would have been readily recognized to one of ordinary skill in the art at the time of the invention that the jump type field would be added to the indirect branch message without affecting the direct branch message" (emphasis added). It is not clear why one of skill in the art would include "the jump type field to the indirect branch message without affecting the direct branch message," as suggested in the Office Action. It is not clear why one of skill in the art would add the field indicating an occurrence of the implicit jump to the Indirect Branch Message of Nexus (shown in Table 6-7) only rather than to, for example, both the Indirect Branch Message and the Direct Branch Message of Nexus (shown in Table 6-6), because the Indirect Branch Message "already" includes the branch target address field.

It appears that the Office Action suggests that one of skill in the art would recognize adding a field indicating a type of the implicit jump to the Indirect Branch Message because the Indirect Branch Message includes the branch target address field. However, none of the cited references supports such suggestion. In contrast, Argade discusses that the trace recording hardware receives, *via an instruction type line, data indicative of instruction types* executed by

the processor core and also receives, *via an inter-module bus, data indicative of program addresses* corresponding to the instruction types received via the instruction type bus (col. 3, lines 35-40) (emphasis added). The trace buffer control *analyzes the stream of instruction types* and corresponding addresses received from the processor core and applies an abbreviation scheme for address information of when a particular instruction type is identified as one of the at least three pre-defined instruction types (Argade, col. 3, lines 51-55) (emphasis added). In addition, the trace recording hardware is capable of identifying whether a particular instruction type was actually executed by the processor core (Argade, col. 3, lines 56-58). Therefore, in Argade, instruction types are analyzed by the trace buffer and are used to provide a program trace. Nowhere does Argade even mention anything that would lead one of skill in the art to realize "that the jump type field would be added to the indirect branch message without affecting the direct branch message," as stated in the Office Action.

In addition, Argade is directed to providing an abbreviated real-time program trace, *containing minimum data necessary to reconstruct a full program trace* (Abstract) (emphasis added). Thus, while Argade is directed to providing the minimum data necessary to reconstruct a full program trace, the reference nevertheless provides INSTR_TYPE signals for **each** instruction execution (emphasis added). Therefore, Argade teaches away from "adding a field to the implicit jump message, the field comprising a second set of bits identifying a type of the implicit jump from among several types of implicit jumps, wherein the field is added only when the address of the destination instruction is not explicitly indicated in the jump instruction," as recited in claim 1.

Accordingly, claim 1 patentably distinguishes over Nexus and Argade, either alone or in combination.

Claims 2-6 depend from claim 1 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 1-6 is respectfully requested.

B. Independent Claim 7

Claim 7, as amended, recites:

> A device for transmitting digital messages between a monitoring circuit integrated on a microprocessor and an analysis tool via output terminals, comprising:

means for detection of a jump on execution of an instruction sequence by the microprocessor;

means for storing data characteristic of the detected jump;

means for generating at least one digital message based on the stored characteristic data, the at least one digital message comprising a first set of bits, wherein:

the first set of bits is set to a first value when the jump is associated with a jump instruction of the instruction sequence explicitly indicating an address of a destination instruction of the jump to provide an explicit jump message, and

the first set of bits set to a second value when the jump is associated with a jump instruction of the instruction sequence not explicitly indicating the address of the destination instruction to provide an implicit jump message indicating an occurrence of an implicit jump; and

means for transmitting the generated at least one digital message;

wherein, *only when the first set of bits is set to the second value, the generation means adds a field to the implicit jump message, the field comprising a second set of bits identifying a type of the implicit jump from among several implicit jump types.*

(Emphasis added).

Claim 7, as amended, recites, inter alia, "only when the first set of bits is set to the second value, the generation means adds a field to the implicit jump message, the field comprising a second set of bits identifying a type of the implicit jump from among several implicit jump type." As discussed above, neither Nexus nor Argade teaches or suggests this limitation.

Accordingly, claim 7 patentably distinguishes over Nexus and Argade, either alone or in combination.

Accordingly, withdrawal of the rejection of claim 7 is respectfully requested.

C. Independent Claim 8

Claim 8, as amended, recites:

A method for transmitting digital messages on execution of an instruction sequence by a microprocessor, the method comprising:

detecting a jump in the execution of the instruction sequence from an initial instruction to a jump destination instruction, wherein the jump destination instruction is different from an instruction following the initial instruction in the instruction sequence;

generating at least one digital message upon the detection of the jump, wherein

only when the jump is implicit, generating the at least one digital message as an implicit jump message indicating an occurrence of an implicit jump and adding an additional field to the implicit jump message, wherein the additional field includes a value identifying a type of the implicit jump, and

when the jump is not implicit, generating the at least one digital message as an explicit jump message; and

transmitting the at least one digital message.

(Emphasis added).

Claim 8, as amended, recites, inter alia, "only when the jump is implicit, generating the at least one digital message as an implicit jump message indicating an occurrence of an implicit jump and adding an additional field to the implicit jump message, wherein the additional field includes a value identifying a type of the implicit jump." As discussed above, neither Nexus nor Argade teaches or suggests this limitation.

Accordingly, claim 8 patentably distinguishes over Nexus and Argade, either alone or in combination.

Claims 9 and 10 depend from claim 8 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 8-10 is respectfully requested.

D. Independent Claim 11

Claim 11, as amended, recites:

A device for transmitting digital messages to monitor operation of a microprocessor, the device comprising:

a monitoring circuit integrated on a microprocessor for:

detecting, on execution of an instruction sequence by the microprocessor, a jump from an initial instruction to a jump destination instruction, wherein the jump destination instruction is different from an instruction following the initial instruction in the instruction sequence;

only when the jump is implicit, adding a field to at least one digital message to provide the at least one digital message as an implicit jump message transmitted on the execution of the instruction sequence by the microprocessor and indicating an occurrence of an implicit jump, wherein the field includes a value identifying a type of the implicit jump; and

when the jump is not implicit, providing the at least one digital message as an explicit jump message;

an analysis tool to reconstitute the instruction sequence based on the at least one digital message; and

at least one monitoring terminal to provide the at least one digital message from the monitoring circuit to the analysis tool.

(Emphasis added).

Claim 11, as amended, recites, inter alia, "only when the jump is implicit, adding a field to at least one digital message to provide the at least one digital message as an implicit jump message transmitted on the execution of the instruction sequence by the microprocessor and indicating an occurrence of an implicit jump, wherein the field includes a value identifying a type of the implicit jump." As discussed above, neither Nexus nor Argade teaches or suggests this limitation.

Accordingly, claim 11 patentably distinguishes over Nexus and Argade, either alone or in combination.

Accordingly, withdrawal of the rejection of claim 11 is respectfully requested.

New Claims

New claims 12-15 have been added to further define Applicants' contribution to the art. Support for these claims can be found at least on page 8, lines 26-29 of Applicants' specification.

Claim 12 depends from claim 1 and is allowable for at least the same reasons.

Claim 13 depends from claim 7 and is allowable for at least the same reasons.

Claim 14 depends from claim 8 and is allowable for at least the same reasons.

Claim 15 depends from claim 11 and is allowable for at least the same reasons.

CONCLUSION

A Notice of Allowance is respectfully requested. The Examiner is requested to call the undersigned at the telephone number listed below if this communication does not place the case in condition for allowance.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicants hereby request any necessary extension of time. If there is a fee occasioned by this response, including an extension fee, the Director is hereby authorized to charge any deficiency or credit any overpayment in the fees filed, asserted to be filed or which should have been filed herewith to our Deposit Account No. 23/2825, under Docket No. S1022.81242US00.

Dated: October 23, 2008

Respectfully submitted,

By: _____

James H. Morris, Reg. No. 34,681
Wolf, Greenfield & Sacks, P.C.
Federal Reserve Plaza
600 Atlantic Avenue
Boston, Massachusetts 02210-2206
Telephone: (617) 646-8000